**Python Journal**

**Name: Cristian Saldana**

**Date: 11 NOV 2022**

**Final Replit Program Share Link:**

[main.py - SpiritedLegitimateBase (1) - Replit](https://replit.com/@CristianCristi5/SpiritedLegitimateBase-1#main.py)

## PART 1: Defining Your Problem

|  |
| --- |
| **Task**  State the problem you are planning to solve.  **Requirements**   * Describe any input data you expect to use. * Describe what the program will do to solve the problem. * Describe any outputs or results the program will provide.   **Inspiration**  When writing your entry below ask yourself the following questions:   * Why do you want to solve this particular problem? * What source(s) of data do you believe you will need? Will the user need to supply that data, or will you get it from an external file or another source? * Will you need to interact with the user throughout the program? Will users continually need to enter data in and see something to continue? * What are your expected results or what will be the end product? What will you need to tell a user of your program when it is complete? |
| 1. Task: To filter out calculations that are too intensive for the computer to handle. 2. The user will supply the necessary input data. 3. The program will act upon conditional statements. 4. The program will decide whether the integers calculated are greater than 100 before proceeding to operate and provide output data. 5. The program will tell the user if there is an error with their input. 6. The program will ask the user if they want to calculate again. |

## PART 2: Working Through Specific Examples

|  |
| --- |
| **Task**  Write down clear and specific steps to solve a simple version of your problem you identified in Part 1.  **Requirements**  Complete the three steps below **for at least two distinct examples/scenarios**.   * State any necessary input data for your simplified problem. * Write clear and specific steps in English (not Python) detailing what the program will do to solve the problem. * Describe the specific result of your example/scenario.   **Inspiration**  When writing your entry below ask yourself the following questions:   * Are there any steps that you don’t fully understand? These are places to spend more time working out the details. Consider adding additional smaller steps in these spots. * Remember that a computer program is very literal. Are there any steps that are unclear? Try giving the steps of your example/scenario to a friend or family member to read through and ask you questions about parts they don’t understand. Rewrite these parts as clearly as you can. * Are there interesting edge cases for your program? Try to start one of your examples/scenarios with input that matches this edge case. How does it change how your program might work? |
| ­Task 1:   1. Type an operation for the calculator to complete (+, -, \*, /). 2. Input two integers less than 100. 3. I pick +, and 1, 1. 4. The calculator will execute since I have picked a valid operation and integers. 5. The result of this scenario is 1+1=2.   Task 2:   1. Type an operation for the calculator to complete (+, -, \*, /). 2. Input two integers less than 100. 3. I pick -, and 2, 2. 4. The calculator will execute since I have picked a valid operation and integers. 5. The result of this scenario is 2-2=0.   Task 3:   1. Type an operation for the calculator to complete (+, -, \*, /). 2. Input two integers less than 100. 3. I pick p, and q, z. 4. The calculator will not execute since I have not picked a valid operation or integers. 5. The result of this scenario is an error and asking to run the program again. |

## PART 3: Generalizing Into Pseudocode

|  |
| --- |
| **Task**  Write out the general sequence your program will use, including all specific examples/scenarios you provided in Part 2.  **Requirements**   * Write pseudocode for the program in English but refer to Python program elements where they are appropriate. The pseudocode should represent the full functionality of the program, not just a simplified version. Pseudocode is broken down enough that the details of the program are no longer in any paragraph form. One statement per line is ideal.   **Help with writing pseudocode**   * Here are a few links that can help you write pseudocode with examples. Remember to check out part 3 of the Example Journal Template Submission if you have not already. Note: everyone will write pseudocode differently. There is no right or wrong way to write it other than to make sure you write it clearly and in as much detail as you can so that it should be easy to convert it to code later.   + <https://www.geeksforgeeks.org/how-to-write-a-pseudo-code/>   + <https://www.wikihow.com/Write-Pseudocode>   **Inspiration**  When writing your entry below ask yourself the following questions:   * Do you see common program elements and patterns in your specific examples/scenarios in Part 2, like variables, conditionals, functions, loops, and classes? These should be part of your pseudocode for the general sequence as well. * Are there places where the steps for your examples/scenarios in Part 2 diverged? These may be places where errors may occur later in the project. Make note of them. * When you are finished with your pseudocode, does it make sense, even to a person that does not know Python? Aim for the clearest description of the steps, as this will make it easier to convert into program code later. |
| This program is a calculator for input integers less than 100.  Create function welcome.  Output “the below 100 calculator, only input integers below 100”.  Initialize welcome function.  Create function calculate.  Output type of operation to be completed.  Obtain type of operation.  Prompt user for 2 integers to be input by passing a string.  Create conditional statement to validate input received is less than 100.  Create conditional statement to validate operation to be completed.  Create conditional statement to initialize error if letters, symbols, non-integers are input.  Create conditional statement to run the program again if error initializes.  Create function again.  Define calc\_again is Y or N.  Create conditional statement for Y and N.  Create conditional statement for N is “see you later”.  Validate operation.  Execute welcome.  Validate input and convert to integer data type by using int function.  Execute calculate.  Validate again\_input.  Execute again.  Execute calculate. |
|  |

## PART 4: Testing Your Program

|  |
| --- |
| **Task**  While writing and testing your program code, describe your tests, record any errors, and state your approach to fixing the errors.  **Requirements**   * For at least one of your test cases, describe how your choices for the test helped you understand whether the program was running correctly or not.   For each error that occurs while writing and testing your code:   * Record the details of the error from Replit. A screenshot or copy-and-paste of the text into the journal entry is acceptable. * Describe what you attempted in order to fix the error. Clearly identify what approach was the one that worked.   **Inspiration**  When writing your entry below ask yourself the following questions:   * Have you tested edge cases and special cases for the inputs of your program code? Often these unexpected values can cause errors in the operation of your program. * Have you tested opportunities for user error? If a user is asked to provide an input, what happens when they give the wrong type of input, like a letter instead of a number, or vice versa? * Did the outcome look the way you expected? Was it formatted correctly? * Does your output align with the solution to the problem you coded for? |
| <Record your errors and fixes here>   1. An error I had was creating a conditional statement for only allowing integers below 100. 2. I was writing an operation to execute instead of creating a conditional statement. 3. The way I fixed the error was to define number\_1 or number\_2 as only accepting integers less than 100 first then allow operations to take place. |

## PART 5: Commenting Your Program

|  |
| --- |
| **Task**  Submit your full program code, including thorough comments describing what each portion of the program should do when working correctly.  **Requirements**   * The purpose of the program and each of its parts should be clear to a reader that does not know the Python programming language.   **Inspiration**  When writing your entry, you are encouraged to consider the following:   * Is each section or sub-section of your code commented to describe what the code is doing? * Give your code with comments to a friend or family member to review. Add additional comments to spots that confuse them to make it clearer. |
| def welcome():  print('''  The below 100 Calculator,  ONLY INPUT INTEGERS BELOW 100  ''')  ...  # Start the call function  welcome()  # Define what calculations the calculator will offer  def calculate():  operation = input('''  Please type in the math operation you would like to complete:  + for addition  - for subtraction  \* for multiplication  / for division  ''')  # This calculator will only ask for two integers  number\_1 = int(input('Please enter the first number: '))  number\_2 = int(input('Please enter the second number: '))    # We start with an if statement to define our paramaters of only accepting integers less than 100    if number\_1 > 100 or number\_2 > 100:  print("Calculator accepts values less than 100 only")    # elif is used hereafter to define other operations we wish to offer in the calculator    elif operation == '+':  print('{} + {} = '.format(number\_1, number\_2))  print(number\_1 + number\_2)  elif operation == '-':  print('{} - {} = '.format(number\_1, number\_2))  print(number\_1 - number\_2)  elif operation == '\*':  print('{} \* {} = '.format(number\_1, number\_2))  print(number\_1 \* number\_2)  elif operation == '/':  print('{} / {} = '.format(number\_1, number\_2))  print(number\_1 / number\_2)  else:  print('You have not typed a valid integer, please run the program again.')  # Add again() function to calculate() function  again()  # start again function  def again():  calc\_again = input('''  Do you want to calculate again?  Please type Y for YES or N for NO.  ''')  if calc\_again.upper() == 'Y':  calculate()  elif calc\_again.upper() == 'N':  print('See you later.')  else:  again()  calculate() |

## PART 6: Your Completed Program

|  |
| --- |
| **Task**  Provide the Replit link to your full program code.  **Requirements**   * The program must work correctly with all the comments included in the program.   **Inspiration**   * Check before submitting your touchstone that your final version of the program is running successfully. |
| [main.py - SpiritedLegitimateBase (1) - Replit](https://replit.com/@CristianCristi5/SpiritedLegitimateBase-1#main.py) |